PARALLEL MULTIPLICATION IN $\mathbb{F}_{2^n}$ USING CONDENSED MATRIX REPRESENTATION

Christophe Negre
Équipe DALI, LP2A, Université de Perpignan
avenue P. Alduy, 66 000 Perpignan, France

Keywords: Finite field, multiplication, matrix representation, irreducible trinomial.

Abstract: In this paper we explore a matrix representation of binary fields $\mathbb{F}_{2^n}$ defined by an irreducible trinomial $P = X^n + X^k + 1$. We obtain a multiplier with time complexity of $T_A + (\lceil \log_2(n) \rceil) T_X$ and space complexity of $(2n-1)n$ AND and $(2n-1)(n-1)$ XOR . This multiplier reaches the lower bound on time complexity. Until now this was possible only for binary field defined by AOP (Silverman, 1999), which are quite few. The interest of this multiplier remains theoretical since the size of the architecture is roughly two times bigger than usual polynomial basis multiplier (Mastrovito, 1991; Koc and Sunar, 1999).

1 INTRODUCTION

A binary field $\mathbb{F}_{2^n} = \mathbb{F}_2[X]/(P)$ is a set of 2^n elements in which we can do all the basic arithmetic operation like addition, subtraction, multiplication and inversion modulo an irreducible binary polynomial P. Finite field arithmetic is widely used in cryptographic applications (Miller, 1985) and error-correcting code (Berlekamp, 1982). For these applications, the most important finite field operation is the multiplication.

The representation of binary field elements have a big influence on the efficiency of field arithmetic. Until now, field elements were represented as sum of basis elements: the basis is composed by n elements $B_1, \ldots, B_n \in \mathbb{F}_{2^n}$, in this situation an element U in $\mathbb{F}_{2^n}$ is written as $U = u_1 B_1, + \cdots + u_n B_n$ with $u_i \in \{0, 1\}$.

The most used bases are polynomial bases (Mastrovito, 1991; Koc and Sunar, 1999; Chang et al., 2005) and normal bases (Wu and Hasan, 1998; Koc and Sunar, 2001).

Our purpose here is to investigate a new representation: the matrix representation. We will focus on field defined by a trinomial $\mathbb{F}_{2^n} = \mathbb{F}_2[X]/(P)$ with $P = X^n + X^k + 1$. In the matrix representation an element U of $\mathbb{F}_{2^n}$ is represented by the n^2 coefficients of a $n \times n$ matrix M_U. The additions of two elements U and V simply consists to add the two matrices M_U and M_V and to multiply U and V it consists to multiply the matrix product $M_U \cdot M_V$.

This gives a faster multiplication than multiplication using basis representation: a parallel multiplier associated to a matrix representation has a time complexity of $T_A + (\lceil \log_2(n) \rceil) T_X$, whereas in basis representation, for field defined by a trinomial (Koc and Sunar, 1999; Mastrovito, 1991), the time complexity is generally equal to $T_A + (2 + \lceil \log_2(n) \rceil) T_X$. The major drawback of this method is due to the length of the representation which requires n^2 coefficients, and provides parallel multiplier with a cubic space complexity in n. But if we carefully select a subset of the matrix coefficients, the number of distinct coefficients in each matrix M_U becomes small: in our situation it is equal to $(2n-1)$. In other words we condense the matrix representation in $(2n-1)$ distinct coefficients to decrease the space complexity.

The paper is organized as follows : in the first section we recall the method of Koc and Sunar (Koc and Sunar, 1999) for finite field multiplication modulo trinomial. They perform the reduction modulo the trinomial on a matrix and then compute a matrix-vector to get the product of two elements. In the second section we study the possibility to use the matrix constructed with Koc and Sunar's method to represent finite field elements. After that we evaluate the complexity of a parallel multiplier in this matrix representation. Next, we study a condensed matrix representation and the associated multiplier. We finally give a small example of our matrix multiplier and finish by a complexity comparison and a brief conclusion.

Negre C. (2006).
PARALLEL MULTIPLICATION IN F2n USING CONDENSED MATRIX REPRESENTATION.
In *Proceedings of the International Conference on Security and Cryptography*, pages 254-259
DOI: 10.5220/0002096402540259

2 MATRIX REPRESENTATION IN POLYNOMIAL BASIS

Let $P = X^n + X^k + 1$ be an irreducible polynomial in $\mathbb{F}_2[X]$. Without loss of generality we can assume $k \leq \frac{n}{2}$ since when $X^n + X^k + 1$ is irreducible, the reciprocal polynomial $X^n + X^{n-k} + 1$ is also irreducible. Furthermore, in this paper we will always suppose for simplicity $k \geq 2$.

An element $U \in \mathbb{F}_{2^n} = \mathbb{F}_2[X]/(P)$ is a polynomial of degree $n-1$. To compute the product of two elements U and V in $\mathbb{F}_{2^n}$ we first compute the product of polynomial

$$W = UV = \left(\sum_{i=0}^{n-1} u_i X^i\right)\left(\sum_{i=0}^{n-1} v_i X^i\right). \qquad (1)$$

This product can be done by a matrix vector product $N_U \cdot V$ where N_U is given below

$$\begin{array}{c} 1 \rightarrow \\ X \rightarrow \\ \vdots \\ X^{n-2} \rightarrow \\ X^{n-1} \rightarrow \\ X^n \rightarrow \\ X^{n-1} \rightarrow \\ \vdots \\ X^{2n-2} \rightarrow \\ X^{2n-1} \rightarrow \end{array} \begin{bmatrix} u_0 & 0 & \cdots & 0 & 0 \\ u_1 & u_0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ u_{n-2} & u_{n-3} & \cdots & u_0 & 0 \\ u_{n-1} & u_{n-2} & \cdots & u_1 & u_0 \\ 0 & u_{n-1} & \cdots & u_2 & u_1 \\ 0 & 0 & \cdots & u_3 & u_2 \\ \vdots & \vdots & \ddots & & \vdots \\ 0 & 0 & \cdots & 0 & u_{n-1} \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}.$$

The product $W = UV$ contains monomials X^i with larger degree than n, i.e., with $n \leq i \leq 2n-2$. These monomials must be reduced modulo $P = X^n + X^k + 1$. To perform this reduction we will use the following identity modulo P for each $i \geq n$

$$X^i = X^{i-(n-k)} + X^{i-n} \mod P. \qquad (2)$$

For example if $P = X^5 + X^2 + 1$ then we have $X^5 = X^2 + 1 \mod P$ and in the same way $X^6 = X^3 + X \mod P$ and so on.

Koc and Sunar in (Koc and Sunar, 1999) have proposed to perform the reduction modulo P on the line of the matrix N_U instead of performing the reduction on the polynomial $W = UV$.

To describe this reducing process we need to state some notations. If M is a $n \times 2n$ matrix, we denote by $(M)_t$ the top part of the matrix M constituted by the n first lines. We will denote also $(M)_l$ the matrix constituted by the n last lines. And finally, we will denote $M[\uparrow s]$ the matrix shifted up by s rows from M, and $M[\downarrow s]$ the matrix shifted down by s rows.

From equation (2) the line corresponding to X^i for $i \geq n$ are pushed up to the lines corresponding to X^{i-n} and X^{i-n+k}. Using the previous notation, this procedure modifies the matrix N_U as follows

$$(N_U)_t \leftarrow (N_U)_t + (N_U)_l + (N_U)_l[\downarrow k] \quad (3)$$
$$(N_U)_l \leftarrow (N_U)_l[\uparrow (n-k)] \quad (4)$$

If we denote by S the low part of N_U, and by T the top part of N_U

$$S = \begin{bmatrix} 0 & u_{n-1} & u_{n-2} & \cdots & u_1 \\ 0 & 0 & u_{n-1} & \cdots & u_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & u_{n-1} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, T = \begin{bmatrix} u_0 & 0 & \cdots & 0 \\ u_1 & u_0 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ u_{n-1} & u_{n-2} & \cdots & u_0 \end{bmatrix},$$

we can rewrite equation (3) and (4) as

$$\begin{aligned} (N_U)_t &= T + S + S[\downarrow k], \\ (N_U)_l &= S[\uparrow (n-k)]. \end{aligned} \qquad (5)$$

Now, since we assumed $k \geq 2$, in the new expression of N_U, the lines corresponding to $X^n, \ldots, X^{n+k-2}$ contains non-zero coefficients. Thus, we have to reduce a second time N_U with the same method. We set $S' = S[\uparrow (n-k)] = (N_U)_l$ and the second reduction provides

$$\begin{aligned} (N_U)_t &= T + S + S' + (S + S')[\downarrow k] \\ (N_U)_l &= 0. \end{aligned} \qquad (6)$$

We finally have the expression of $M_U = (N_U)_t$ the reduced form of N_U

$$M_U = \begin{bmatrix} u_0 & u_{n-1} & \cdots & \cdots & \cdots & u_k & u'_{k-1} & \cdots & \cdots & u'_1 \\ u_1 & u_0 & u_{n-1} & \cdots & \cdots & u_k & u'_{k-1} & \cdots & & u'_2 \\ \vdots & & \ddots & \ddots & & & \ddots & \ddots & & \vdots \\ u_{k-2} & \cdots & \cdots & u_0 & u_{n-1} & \cdots & \cdots & u_k & u'_{k-1} & \\ u_{k-1} & u_{k-2} & \cdots & \cdots & u_0 & u_{n-1} & \cdots & \cdots & u_k & \\ u_k & u'_{k-1} & u'_{k-2} & \cdots & \cdots & u'_0 & u''_{n-1} & \cdots & \cdots & u''_{k+1} \\ u_{k+1} & u_k & u'_{k-1} & \cdots & \cdots & \cdots & u'_0 & u''_{n-1} & \cdots & u''_{k+2} \\ \vdots & & \ddots & \ddots & & & & \ddots & \ddots & \vdots \\ u_{n-2} & \cdots & \cdots & u_k & u'_{k-1} & \cdots & \cdots & \cdots & u'_0 & u''_{n-1} \\ u_{n-1} & \cdots & \cdots & \cdots & u_k & u'_{k-1} & \cdots & \cdots & \cdots & u'_0 \end{bmatrix}, \qquad (7)$$

where

$$u'_i = u_i + u_{i+(n-k)} \text{ for } 0 \leq i < k,$$

$$u''_i = \begin{cases} u_i + u_{i-k} + u_{i+n-2k} \text{ for } k \leq i < 2k, \\ u_i + u_{i-k} \text{ for } 2k \leq i < n. \end{cases}$$

The method of Koc and Sunar (Koc and Sunar, 1999) to compute the product $UV \mod P$, first consists to compute the coefficients u'_i and u''_i of M_U and after that to perform the matrix-vector product $M_U \cdot V$ to obtain $UV \mod P$. This multiplier computes the product in time $T_A + (\lceil \log_2(n) \rceil + 2)T_X$ using a parallel architecture.

If we know the coefficients u_i, u'_i and u''_i we avoid the delay to compute these coefficients. In this situation the product could be done in time $T_A +$

$\lceil \log_2(n) \rceil T_X$. This remark pushed us to try to keep the field elements $U \in \mathbb{F}_{2^n}$ expressed by the matrix M_U (i.e., in this case we always know the coefficients u_i, u'_i and u''_i, and we don't have to compute it before the multiplication) and try to use this representation to implement finite field arithmetic.

Definition (Matrix Representation)**.** *Let $\mathbb{F}_{2^n} = \mathbb{F}_2[X]/(P)$ where $P = X^n + X^k + 1$ with $2 \leq k \leq \frac{n}{2}$. The matrix representation of an element $U = \sum_{i=0}^{n-1} u_i X^i$ of the field $\mathbb{F}_{2^n}$ is the matrix given in equation* (7) *expressed in term of the coefficients u_i, u'_i and u''_i.*

The next section is devoted to explain how to add and multiply field elements in matrix representation.

3 FIELD ARITHMETIC IN MATRIX REPRESENTATION

Let $\mathbb{F}_{2^n} = \mathbb{F}_2[X]/(P)$ where P is an irreducible trinomial $P = X^n + X^k + 1$ with $2 \leq k \leq \frac{n}{2}$.

The following Theorem shows that, if the elements $U \in \mathbb{F}_{2^n}$ are represented by their associated matrix M_U, finite field arithmetic corresponds to classical $n \times n$ matrix arithmetic.

Theorem 1. *Let $\mathbb{F}_{2^n} = \mathbb{F}_2[X]/(P)$ where P is an irreducible trinomial $P = X^n + X^k + 1$ with $2 \leq k \leq \frac{n}{2}$. Let $U = \sum_{i=0}^{n-1} u_i X^i$ and $V = \sum_{i=0}^{n-1} v_i X^i$ be two elements in $\mathbb{F}_{2^n}$ and M_U and M_V their corresponding matrix defined in* (7)*. If $W_1 = U + V \mod P$ and $W_2 = UV \mod P$ we have*

$$M_{W_1} = M_U + M_V, \quad (8)$$
$$M_{W_2} = M_U \cdot M_V. \quad (9)$$

Proof. Using equation 7, it is clear that to show the assertion for W_1, it is sufficient to show that

$$\begin{aligned} w_i &= u_i + v_i, \\ w'_i &= u'_i + v'_i, \\ w''_i &= u''_i + v''_i. \end{aligned}$$

The identity on w_i is trivial since $W_1 = U + V \mod P$. For w'_i we have

$$w'_i = w_i + w_{i+(n-k)} = (u_i + v_i) + (u_{i+(n-k)} + v_{i+(n-k)}).$$

By rearranging this expression, we get

$$w'_i = (u_i + u_{i+(n-k)}) + (v_i + v_{i+(n-k)}) = u'_i + v'_i.$$

A similar proof can be done to show that $w''_i = u''_i + v''_i$.

For the assertion on M_{W_2} it is a little bit more difficult. We remark that from the result of section 2, for every elements $Z \in \mathbb{F}_{2^n}$ the product VZ in $\mathbb{F}_{2^n}$ is given by $M_V \cdot Z$ and the product $W_2 Z$ by $M_{W_2} \cdot Z$. Thus we get for every $Z \in \mathbb{F}_{2^n}$ that

$$\begin{aligned} M_{W_2} \cdot Z &= W_2 Z = UVZ \\ &= M_U \cdot (VZ) = M_U \cdot (M_V \cdot Z) \\ &= (M_U \cdot M_V) \cdot Z \end{aligned}$$

This implies that $(M_W - M_U M_V) \cdot Z = 0$ for each Z in $\mathbb{F}_{2^n}$, but this means that $(M_W - M_U M_V)$ is the zero matrix. In other words we have $M_W = M_U \cdot M_V$ as required. □

For a more general proof see (Lidl and Niederreiter, 1986). The following example illustrates the Theorem 1.

Example 1. We consider the field $\mathbb{F}_{2^7} = \mathbb{F}_2[X]/(X^7 + X^3 + 1)$ and let $U = \sum_{i=0}^{n-1} u_i X^i$ be an element of $\mathbb{F}_{2^7}$. From equation (7) we get the following expression of M_U

$$M_U = \begin{bmatrix} u_0 & u_6 & u_5 & u_4 & u_3 & u'_2 & u'_1 \\ u_1 & u_0 & u_6 & u_5 & u_4 & u_3 & u'_2 \\ u_2 & u_1 & u_0 & u_6 & u_5 & u_4 & u_3 \\ u_3 & u'_2 & u'_1 & u'_0 & u''_6 & u''_5 & u''_4 \\ u_4 & u_3 & u'_2 & u'_1 & u'_0 & u''_6 & u''_5 \\ u_5 & u_4 & u_3 & u'_2 & u'_1 & u'_0 & u''_6 \\ u_6 & u_5 & u_4 & u_3 & u'_2 & u'_1 & u'_0 \end{bmatrix}$$

with $u'_0 = u_0 + u_4, u'_1 = u_1 + u_5, u'_2 = u_2 + u_6$ and $u''_4 = u'_4 + u_1 + u_5, u''_5 = u_5 + u_2 + u_6, u''_6 = u_6 + u_3$.

For $U = 1 + X + X^4$ and $V = X^2 + X^3 + X^5$ we obtain the following matrices

$$M_U = \begin{bmatrix} 1&0&0&1&0&0&1 \\ 1&1&0&0&1&0&0 \\ 0&1&1&0&0&1&0 \\ 0&0&1&0&0&0&0 \\ 1&0&0&1&0&0&0 \\ 0&1&0&0&1&0&0 \\ 0&0&1&0&0&1&0 \end{bmatrix}, \quad M_V = \begin{bmatrix} 0&0&1&0&1&1&1 \\ 0&0&0&1&0&1&1 \\ 1&0&0&0&1&0&1 \\ 1&1&1&0&1&0&1 \\ 0&1&1&1&0&1&0 \\ 1&0&1&1&1&0&1 \\ 0&1&0&1&1&1&0 \end{bmatrix}.$$

Now we add M_U and M_V to get the matrix $M_{W_1} = M_U + M_V$ of $W_1 = U + V = X^5 + X^4 + X^3 + X^3 + X + 1 \mod P$ and we multiply M_U and M_V to get the matrix $M_{W_2} = M_U \cdot M_V$ of $W_2 = UV = X^4 + X^3 + 1 \mod P$

$$M_{W_1} = \begin{bmatrix} 1&0&1&1&1&1&0 \\ 1&1&0&1&1&1&1 \\ 1&1&1&0&1&1&1 \\ 1&1&0&0&1&0&1 \\ 1&1&1&0&0&1&0 \\ 1&1&1&1&0&0&1 \\ 0&1&1&1&1&0&0 \end{bmatrix}, M_{W_2} = \begin{bmatrix} 1&0&0&1&1&0&0 \\ 0&1&0&0&1&1&0 \\ 0&0&1&0&0&1&1 \\ 1&0&0&0&1&0&1 \\ 1&1&0&0&0&1&0 \\ 0&1&1&0&0&0&1 \\ 0&0&1&1&0&0&0 \end{bmatrix}.$$

4 PARALLEL MULTIPLICATION IN MATRIX REPRESENTATION

Let us now study the architecture of the multiplier associated to the matrix representation. We fix $U, V \in$

$\mathbb{F}_{2^n}$ and M_U and M_V their associated matrix. Let $W = UV$ be the product of U and V in $\mathbb{F}_{2^n}$ and M_W its associated matrix . We will note $L_i(M_U)$ for $i = 0, \ldots, n-1$ the line of M_U and $C_j(M_V)$ for $j = 0, \ldots, n-1$ the columns of M_V.

The coefficient $\text{Coeff}_{i,j}(M_W)$ of index (i, j) of M_W is then computed as a line-column matrix product (in the sequel we will call this operation a *scalar product*)

$$\begin{aligned} &\text{Coeff}_{i,j}(M_W) = L_i(M_U) \cdot C_j(M_V) \\ &= \sum_{\ell=0}^{n-1} \text{Coeff}_{i,\ell}(M_U)\text{Coeff}_{\ell,j}(M_V) \end{aligned} \quad (10)$$

A scalar product (10) can be done in time $T_A + \lceil \log_2(n) \rceil T_X$, where T_A is the delay for an AND gate and T_X for an XOR gate, using parallel AND gates and a binary tree of XOR.

Consequently, if all these scalar products are done in parallel, one can compute the product M_W of M_U and M_V in time $T_A + \lceil \log_2(n) \rceil T_X$. So at this point we reach the lower bound on time complexity in binary field multiplication.

The major drawback of this approach is that we have to compute n^2 coefficients $\text{Coeff}_{i,j}(M_W)$. The space complexity is thus roughly n^3 AND and n^3 XOR which is widely too big and not practical.

But in fact, we did not use the fact that the number of distinct coefficients in each matrix M_U, M_V and M_W is quite small. A lot of scalar products can be avoided, this motivates the use of a condensed matrix representation.

5 CONDENSED MATRIX REPRESENTATION

The set of coefficients of the matrix M_U for a given U is quite small: it consists of the n bits u_i, the k bits u'_i and the $(n-k-1)$ bits u''_i. The matrix representation M_U can be condensed in this three set of coefficients.

Definition (Condensed Matrix Representation)**.** *We consider the field $\mathbb{F}_{2^n} = \mathbb{F}_2[X]/(P)$ where P is an irreducible trinomial, and let $U = \sum_{i=0}^{n-1} u_i X^i$ be an element of $\mathbb{F}_{2^n}$. The condensed matrix representation of U is CMR$(U) = (U, U'', U''')$ such that*

$U = (u_0, \ldots, u_{n-1})$,
$U' = (u'_0, \ldots, u'_{k-1})$, *where* $u'_i = u_i + u_{i+(n-k)}$,
$U'' = (u''_i, \ldots, u''_j)$, *where*

- $u''_i = u_i + u_{i-k} + u_{i+n-2k}$ *for* $i = k+1, \ldots, 2k-1$,
- $u''_i = u_i + u_{i-k}$ *for* $i = 2k, \ldots, n-1$.

The condensed matrix representation of U contains all the distinct coefficients of the matrix M_U. We can thus reconstruct each line and each column of matrix M_U.

Construction of the lines of M_U. We note $L_i(M_U)$ the line of M_U for $i = 0, \ldots, n-1$. Using the expression of M_U of equation (7), we get the following expression of these lines of M_U in term of the CMR of U in the Table 1.

Table 1: Lines of M_U.

i=0,...,k−2	$L_i(M_U)$=$\left[u_i u_{i-1} \cdots u_0 u_{n-1} \cdots u_k u'_{k-1} \ldots u'_{i+1}\right]$
i=k−1	$L_{k-1,A}$=$\left[u_{k-1} u_{k-2} \cdots u_0 u_{n-1} \cdots u_k\right]$
i=k,...n−2	$L_i(M_U)$=$\left[u_i \ldots u_k u'_{k-1} \ldots u'_0 u''_{n-1} \ldots u''_{i+1}\right]$
i=n−1	$L_i(M_U)$=$\left[u_{n-1} \ldots u_k u'_{k-1} \ldots u'_0\right]$

Construction of the columns of M_U. We note $C_j(M_U)$ the columns of M_U for $j = 0, \ldots, n-1$. From (7) we get the Table 2 of the columns of M_U where the expression of $C_j(M_U)$ are given in term of the CM representation of U.

Now using these descriptions of the lines and the columns of the matrix M_U we can easily express the multiplication in the condensed matrix representation.

5.1 Multiplication in Condensed Matrix Representation

Let U and V be two elements of $\mathbb{F}_{2^n}$ given by their respective condensed matrix representation CMR$(U) = (U, U', U'')$ and CMR$(V) = (V, V', V'')$. As stated in the previous section, with the CMR representation of U and V we can easily construct the lines and columns of M_U and M_V. We want to compute the coefficients of the condensed matrix representation of the product $W = UV$ in $\mathbb{F}_{2^n} = \mathbb{F}_2[X]/(X^n + X^k + 1)$. To do this first recall that, from Theorem 1, for each $0 \leq i, j \leq n-1$ we have

$$\text{Coeff}_{i,j}(M_W) = L_i(M_U) \cdot C_j(M_V). \quad (11)$$

To get the CMR of W we need only to compute the coefficients W, W', W'' of M_W.

- *Computing the coefficients of W.* The coefficients w_i of CMR(W) are in the first column of M_W. This means that

$$w_j = \text{Coeff}_{0,j}(M_W) = L_0(M_U) \cdot C_j(M_V).$$

- *Computing the coefficients of W' and W''.* If we look at equation (7) we can see that W' and W'' appears in the line $L_k(M_W)$, i.e.,

$$w'_j = \text{Coeff}_{k,j}(M_W) \quad \text{for } 1 \leq j \leq k,$$
$$w''_j = \text{Coeff}_{k,j}(M_W) \quad \text{for } k+1 \leq j \leq n-k.$$

Table 2: The columns of M_U.

$j=0$	$C_0(M_U) = {}^t[u_0 \cdots \cdots \cdots u_{n-1}]$
$j=1,\ldots,k-1$	$C_j(M_U) = {}^t[u_{n-j}\cdots u_{n-1}u_0\cdots u_{k-j-1}u'_{k-j}\cdots u'_{k-1}u_k\cdots u_{n-j-1}]$
$j=k,\ldots,n-k$	$C_j(M_U) = {}^t[u_{n-j}\cdots u_{n-j+k-1}u''_{n-j+k}\cdots u''_{n-1}u'_0\cdots u'_k u_{k+1}\cdots u_{n-j-1}]$
$j=n-k+1,\ldots,n-1$	$C_j(M_U) = {}^t[u'_{n-j}\cdots u'_{k-1}u_k\cdots u_{n-j+k-1}u''_{n-j+k}\cdots u''_{n-1}u'_0\cdots u'_{n-j-1}]$.

Now using (11) we get the following expression for w'_j and w''_j

$$w'_j = L_k(M_U)\cdot C_j(M_V) \quad \text{for } 1 \le j \le k,$$
$$w''_j = L_k(M_U)\cdot C_j(M_V) \quad \text{for } k+1 \le j < n.$$

These operations can be done in a parallel hardware architecture. Specially each coefficient w_i, w'_i and w'_i we perform in parallel a scalar product through parallel AND gates, and a binary tree of XOR.

Complexity. Let us evaluate the complexity of this multiplier. It consists in $(2n-1)$ scalar products done in parallel:

- n for the coefficients $w_j,\ j=0,\ldots,n-1$,
- $n-1$ for the w'_j, with $j=0,\ldots,k-1$ and w''_j with $j=k+1,\ldots,n-1$.

Since one scalar product requires n AND and $(n-1)$ XOR, the overall space complexity of the multiplier is equal to $(2n-1)n$ AND and $(2n-1)(n-1)$ XOR . For the time complexity, since the computation of the coefficients of CMR(W) are done in parallel, the time complexity is equal to the delay of only one scalar product. But this delay is equal to $T_A + \lceil \log_2(n) \rceil T_X$.

6 EXAMPLE

We consider the field $\mathbb{F}_{2^5} = \mathbb{F}_2[X]/(X^5+X^2+1)$. Let $U = u_0 + u_1X + u_2X^2 + u_3X^3 + u_4X^4 \in \mathbb{F}_{2^5}$. The condensed matrix representation of U is given by the three vectors

$$\begin{array}{ll} U & = (u_0,u_1,u_2,u_3,u_4,u_5), \\ U' & = (u'_0,u'_1),\ U'' = (u''_3,u''_4). \end{array}$$

where the u'_i and u''_i are defined by

$$\begin{array}{ll} u'_0 & = u_0+u_3,\ u'_1 = u_1+u_4, \\ u''_3 & = u_3+u_1+u_4,\ u''_4 = u_4+u_2, \end{array}$$

Using the general formula (7) for M_U we get that M_U is as follows

$$M_U = \begin{bmatrix} u_0 & u_4 & u_3 & u_2 & u'_1 \\ u_1 & u_0 & u_4 & u_3 & u_2 \\ u_2 & u'_1 & u'_0 & u''_4 & u''_3 \\ u_3 & u_2 & u'_1 & u'_0 & u''_4 \\ u_4 & u_3 & u_2 & u'_1 & u'_0 \end{bmatrix}.$$

Now if $U = 1+X^3+X^4$ and $V = X+X^2$ their condensed matrix representation are

$$\text{CMR}(U) = \begin{cases} U = (1,0,0,1,1), \\ U' = (0,1), U'' = (0,1). \end{cases}$$
$$\text{CMR}(V) = \begin{cases} V = (0,1,1,0,0), \\ V' = (0,1), V'' = (1,1). \end{cases}$$

Using the vectors U, U', U'' we can construct M_U and with the vectors V, V'' and V'' we can construct M_V

$$M_U = \begin{bmatrix} 1\,1\,1\,0\,1 \\ 0\,1\,1\,1\,0 \\ 0\,1\,0\,1\,0 \\ 1\,0\,1\,0\,1 \\ 1\,1\,0\,1\,0 \end{bmatrix},\ M_V = \begin{bmatrix} 0\,0\,0\,1\,1 \\ 1\,0\,0\,0\,1 \\ 1\,1\,0\,1\,1 \\ 0\,1\,1\,0\,1 \\ 0\,0\,1\,1\,0 \end{bmatrix}.$$

Let $W = UV$ be the product of U and V in $\mathbb{F}_{2^5}$. We compute the condensed matrix representation of W by multiplying well chosen line of M_U and well chosen line of M_V as described in the previous section.

We compute w_0 by multiplying $L_0(M_U)$ the first line of M_U with $C_0(M_V)$ the first column of M_V

$$\begin{array}{rcl} w_0 & = & L_0(M_U)\cdot C_0(M_V) \\ & = & [1\,1\,1\,0\,1]\cdot {}^t[0\,1\,1\,0\,0] = 0 \end{array}$$

For $w_i,\ i=1,2,3,4$ we have

$$\begin{array}{l} w_1 = L_1(M_U)\cdot C_0(M_V) = 0, \\ w_2 = L_2(M_U)\cdot C_0(M_V) = 1, \\ w_3 = L_3(M_U)\cdot C_0(M_V) = 1, \\ w_4 = L_4(M_U)\cdot C_0(M_V) = 1. \end{array}$$

For the coefficients w'_i we do

$$\begin{array}{l} w'_1 = L_2(M_U)\cdot C_1(M_V) = 1, \\ w'_2 = L_2(M_U)\cdot C_2(M_V) = 1. \end{array}$$

And finally for the coefficients w''_i we have

$$\begin{array}{l} w''_3 = L_2(M_U)\cdot C_3(M_V) = 0, \\ w''_4 = L_2(M_U)\cdot C_4(M_V) = 0. \end{array}$$

We can easily check that the coefficients w_i, w'_i and w''_i are the correct coefficients of the condensed matrix representation of the product of U and V in $\mathbb{F}_{2^5}$.

Table 3: Complexity.

Algorithm	Space complexity		Time complexity
	# AND	# XOR	
CMR for $X^n + X^k + 1$ with $2 \leq k \leq \frac{n}{2}$ (this paper)	$(2n-1)n$	$(2n-1)(n-1)$	$T_A + (\lceil \log_2(n) \rceil) T_X$
PB for $X^n + X^k + 1$ with $2 \leq k \leq \frac{n}{2}$ (Koc and Sunar, 1999)	n^2	$(n^2 - k)$	$T_A + (2 + \lceil \log_2(n) \rceil) T_X$
PB for $X^n + X + 1$ (Koc and Sunar, 1999)	n^2	$(n^2 - 1)$	$T_A + (1 + \lceil \log_2(n) \rceil) T_X$
PB for AOP (Chang et al., 2005)	$(\frac{3n^2}{4} + 2n + 1)$	$+\frac{3(n+2)^2}{4}$	$T_A + (1 + \lceil \log_2(n) \rceil) T_X$
NB of Type I (Wu and Hasan, 1998)	n^2	$(n^2 - 1)$	$T_A + (1 + \lceil \log_2(n) \rceil) T_X$
NB of type II (Koc and Sunar, 2001)	n^2	$\frac{3}{2}(n^2 - n)$	$T_A + (1 + \lceil \log_2(n) \rceil) T_X$

7 CONCLUSION

We have presented in this paper a new multiplier architecture for binary field $\mathbb{F}_{2^n}$ generated by a trinomial $X^n + X^k + 1$ with $2 \leq k \leq \frac{n}{2}$ using a condensed matrix representation. This multiplier is highly parallelizable.

In Table 3 we give the complexity of our architecture, and also the complexity of different multiplier architectures proposed in the literature for field $\mathbb{F}_{2^n}$.

We see that the use of a condensed matrix representation provides a multiplication which is the faster among all previously proposed multiplier known by the author. The gain on time complexity for multiplication modulo trinomials $X^n + X^k + 1$ with cryptographic size $n \sim 160$ is around 20% when $k \geq 2$ compared to polynomial basis multiplier and 10% compared to normal basis multiplier or AOP polynomial bases. But we have to pay a big price for this improvement : the condensed matrix representation parallel multiplier has a space complexity which is roughly two times bigger than classical polynomial and normal basis multiplier.

REFERENCES

Berlekamp, E. (1982). Bit-serial Reed-Solomon encoder. *IEEE Trans. Information Theory*, IT-28:869–874.

Chang, K.-Y., Hong, D., and Cho, H.-S. (2005). Low complexity bit-parallel multiplier for GF(2^m) defined by all-one polynomials using redundant representation. *IEEE Trans. Comput.*, 54(12):1628–1630.

Koc, C. and Sunar, B. (1999). Mastrovito Multiplier for All Trinomials. *IEEE Transaction on Computers*, 48(5):522–52.

Koc, C. and Sunar, B. (2001). An Efficient Optimal Normal Basis Type II Multiplier. *IEEE Trans. on Computers*, 50:83–87.

Lidl, R. and Niederreiter, H. (1986). *Introduction to Finite Fields and Their Applications*. Cambridge Univ Press.

Mastrovito, E. (1991). *VLSI architectures for computations in Galois fields*. PhD thesis, Dep.Elec.Eng.,Linkoping Univ.

Miller, V. (1985). Uses of elliptic curves in cryptography. *Advances in Cryptology, proceeding's of CRYPTO'85, Lecture Note in Computer Science 218.*

Silverman, J. H. (1999). Fast Multiplication in Finite Fields GF(2^n). In *Crytographic Hardware and Embedded Systems - CHES'99*, volume 1717 of *LNCS*, pages 122–134.

Wu, H. and Hasan, M. (1998). Low-Complexity Multipliers Bit-Parallel for a Class of Finite Fields. *IEEE Trans. Computers*, 47:883–887.